Monk and Multiplication

Link submit: https://www.hackerearth.com/practice/data-structures/trees/heapspriority-queues/practice-problems/algorithm/monk-and-multiplication/

Solution:

C++	http://ideone.com/mYnqT0
Java	http://ideone.com/sp6bID
Python	http://ideone.com/ZDyJkY

Tóm tắt đề:

Monk đang tìm hiểu hàng đợi ưu tiên. Cho cậu một mảng số nguyên. Với mỗi phần tử khi thêm vào hàng đợi ưu tiên, Monk muốn tìm tích của ba phần tử lớn nhất hiện có trong hàng đợi.

Nếu không có đủ ba số lớn nhất, nhì và ba thì in ra -1.

Input:

Dòng đầu tiên chứa số *N* (1 ≤ N ≤ 100.000) – số lượng phần tử trong mảng.

Dòng tiếp theo chứa N phần tử trong mảng (0 ≤ A[i] ≤ 1.000.000).

Output:

Với mỗi phần tử khi thêm vào thì in ra tích của số lớn nhất, số lớn nhì, số lớn ba. Không có đủ ba số thì in ra -1.

Ví dụ:

5 1 2 3 4 5	-1 -1 6 24 60

Giải thích ví dụ:

Thêm 1 vào hàng đợi ưu tiên → in ra -1.

Thêm 2 vào hàng đợi ưu tiên → in ra -1.

Thêm 3 vào hàng đợi ưu tiên → in ra 3 * 2 * 1 = 6.

Thêm 4 vào hàng đợi ưu tiên → in ra 4 * 3 * 2 = 24.

Thêm 5 vào hàng đợi ưu tiên → in ra 5 * 4 * 3 = 60.

Hướng dẫn giải:

Áp dụng hàng đợi ưu tiên là giải quyết bài này.

- Bước 1: Lưu các phần tử đề cho vào một mảng.
- Bước 2: Lần lượt duyệt qua các phần tử và bỏ vào hàng đợi ưu tiên.
  - Nếu như số lượng phần tử trong hàng đợi ưu tiên nhỏ hơn 3 thì ta xuất ra -1.
  - Mỗi lần bỏ phần tử mới vào thì lấy top 3 để nhân với nhau và in ra kết quả.

Độ phức tạp: **O(NlogN)** với N là số lượng phần tử trong mảng.